REMARKS

The non-final Office Action mailed November 26, 2003 and the references cited therein have been carefully considered. The specification and Claims 15, 16, 45, and 46 have been amended in a sincere effort to further clarify that which Applicant regards as the invention. Support for this amendment is found generally within the specification, claims, and drawings, as originally filed.

In the Office Action, the Examiner rejected Claims 15-30 and 45-60 under 35 U.S.C. §112, second paragraph, as failing to comply with the enablement requirement. Specifically, the Examiner indicates that recitation of the term "operational plane memory" in Claims 15, 16, 45, and 46 is subject matter that was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or to which it is most nearly connected, to make and/or use the invention. The Examiner then cited page 4, lines 16-17 as mentioning the term "operational plane memory", but states that the specification does not describe what this type of memory is and that the difference between "operational plane memory" and "memory" is unclear.

Applicant respectfully refers the Examiner to references to operational plane memory throughout the specification, including at page 4, lines 18-23; page 8, lines 6 through page 9, line 17; and page 10, lines 6-19. In addition, "table memory 56 preferably stores each of the tables to be searched and operational plane memory preferably stores the particular table currently being searched", which is defined at page 8, lines 21-23 of the specification. Therefore it is respectfully requested that the rejection of Claims 15-30 and 45-60 under 35 U.S.C. §112, first paragraph, be reconsidered and withdrawn.

Claims 15-30 and 45-60 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Publication No. 2002/0111699 to Melli et al. (*Melli*) in view of U.S. Patent No.

6,625,592 to Dyer et al. (*Dyer*). Specifically, with respect to Claims 15, 16, 45, and 46, the Examiner indicates that *Melli* discloses each of the recited elements, except for transferring substantially simultaneously the table in parallel from table memory to operational plane memory in the integrated circuit and outputting a result of the search function. However, the Examiner indicates that *Dyer* teaches these elements at column 5, lines 4-46. The Examiner further states that it would have been obvious to modify *Melli* to include the feature disclosed in *Dyer*, since doing so allows for faster processing and more efficient use of an integrated circuit in a computer.

The present invention is directed to a method of performing a search function in an integrated circuit, which includes the steps of storing a table in a table memory in the integrated circuit, inputting a search key, and transferring substantially simultaneously the table in parallel from the table memory to an operational plain memory in the integrated circuit. The method also includes performing at least one search function on the table in the operational plane memory using a search key, and outputting a result of the search function, as now defined by amended Claim 15.

The present invention is also directed to a method of performing a search function in an integrated circuit, which includes the steps of storing a plurality of tables in a table memory in the integrated circuit, inputting a table identifier, and inputting a search key. The method also includes transferring substantially simultaneously at least one of the plurality of tables represented by the table identifier in parallel from the table memory to an operational plane memory in the integrated circuit, performing at least one search function on the at least one table in the operational plane memory using the search key, and outputting result of the search function, as now defined by amended Claim 16.

The present invention is further directed to a method of performing a search function, which includes the steps of inputting an unsorted entry, and performing a hash function on

the unsorted entries, wherein the hash function arranges the unsorted entries into a sorted table. The method further includes storing the sorted table in a table memory in an integrated circuit, inputting a search key, and transferring substantially simultaneously the sorted table in parallel from the table memory to an operational plane memory in the integrated circuit. The method also includes performing at least one search function on the sorted table in the operational plane memory using the search key, and outputting the result of the search function, as now defined by amended Claim 45.

The present invention is yet further directed to a method of performing a search function, which includes the steps of inputting unsorted entries, and performing a first hash function on the unsorted entries, wherein the first hash function arranges the unsorted entries into a plurality of sorted tables. The method also includes storing the plurality of sorted tables in a table memory in an integrated circuit, inputting a search key, and performing a second hash function on the search key, wherein the second hash function outputs a table identifier representative one of the plurality of sorted tables in which the search key is likely to be found. The method further includes transferring substantially simultaneously at least one of the plurality of tables represented by the table identifier in parallel from the table memory to an operational plane memory in the integrated circuit, performing at least one search function on the at least one table in the operational plane memory using the search key, and outputting a result of the search function, as now defined by amended Claim 46.

Melli relates to a method and system for creating and using a generic container as a data structure in a software program. A selected data structure is specified at startup or run-time using a mapping table that tracks specified data structures for each generic container used by the program. Each data structure is abstracted to a generic interface for use with the container. The program interacts with the generic container using the generic interface, which allows the container to add, delete, and lookup data stored in the container, as well as

to retrieve a key used by the container. The program uses an object factory to create an instance of a generic container having the generic interface.

At run-time, the object factory refers to the mapping table to determine which data structure is used for each generic container, as described at page 1, paragraphs 5, 12, and 13. However, nothing in *Melli* would teach or suggest performing a search on a table located in a separate area (operational plane memory) from that in which it was originally stored (table memory) as now defined by amended Claims 15, 16, 45 , and 46. In addition, as the Examiner concedes, *Melli* does not teach the substantially simultaneous transfer of the table from operational plane memory to table memory, as now defined by amended Claims 15, 16, 45, and 46.

Dyer relates to a system, method, and data structure delineated for use in data storage and retrieval. Records stored in shared memory contain data entries and associated hash codes computed from a predetermined hash function. The search request uses the same hash function to generate user codes. The records are searched to find one with hash codes that equal the user codes. This search involves comparisons of native data type entries that are intended to be faster than character string comparisons. After a record is identified with hash codes matching the user codes, a more comprehensive comparison is made between the selected record and respective data entries of the search request, as described at column 3, line 8 through column 4, line 61.

Although the Examiner states that column 4, line 62 through column 5, line 46 describe substantially simultaneously transferring a table in parallel from the table memory to an operational plane memory and outputting a result of the search function, it is respectfully submitted that this portion of the reference merely describes the initial entry of data in step 24 as a single step 26 shown in Figure 4, and that the generation of the hash code in step 28 may be carried out in parallel, if desired. Specifically, column 5, lines 14-20 state the following:

> In this example of process 24 operation, all data entries are input in a single step 26, and then sequentially processed for hash code generation in step 28. Those skilled in the art understand that the hash code generation of step 28 could be carried out in parallel, if desired. Moreover, data entry in step 26 could be implemented on a one-field-at-a-time basis, as opposed to entering all fields in one step.

Specifically, *Dyer* states that "all data entries are input in a single step 26, and then sequentially processed for hash code generation in step 28". Firstly, it is respectfully submitted that this refers only to the initial piecemeal entry of data corresponding to a single employee record "using a keyboard, a message from another computer 74 in the network or from a device external to the network" as described at column 4 line 66 through column 5, line 2, and not the transfer of a complete table from table memory to operational plane memory in preparation for performing at least one search function in the operational plane memory using the search key, as now defined by amended Claims 15, 16, 45, and 46. Support for this interpretation is provided at column 5, lines 10-13, which states that the result of steps 26 and 28 is stored in a single record 12a, which refers to a single employee, and that similar iterations of steps 26 and 28 are required for data entry in the remaining individual employee records 12b-e.

Secondly, it is respectfully submitted that initial entry of data in a single step followed by sequentially processing the entered data merely requires that all data for a particular employee be entered before hash code generation begins for that employee. It does not mean that the complete transfer of a table occurs simultaneously from table memory to operational plane memory, as now defined by amended Claims 15, 16, 45, and 46. The single step 26 referred to in *Dyer* merely indicates that no other step is performed before all the data for a

single employee has been entered, not that data for all employees is transferred simultaneously in parallel between separate memories.

Thirdly, *Dyer* states "that those skilled in the art understand that the hash code generation of step 28 could be carried out in parallel". However, it is respectfully submitted that this statement merely refers to the hash code generation step and that this step alone may be performed in parallel. That is, that the three segments of hash code generation 30, 32, and 34 may be performed in parallel on any specified employee record rather than sequentially, as shown in Figure 4.

Fourthly, *Dyer* indicates that "data entry in step 26 could be implemented on a "one-field-at-a-time basis, as opposed to entering all fields in one step". It is respectfully submitted that this means that hash code generation need not wait for a complete employee record to be entered in step 26 before generating hash codes in step 28, but that hash code generation may begin on each field of the employee record as soon as it is entered.

Therefore, nothing in *Dyer* would teach or suggest transferring substantially simultaneously at least one complete table to be searched in parallel from table memory to operational plane memory and then performing a search function on that table in a separate portion of memory, that is, operational plane memory, as now defined by Claims 15, 16, 45, and 46. Further, it is respectfully submitted that nothing in any of the art of record would teach or suggest combining the references in the manner suggested by the Examiner to allow for faster processing and more efficient use of an integrated circuit, and to do so would involve improper hindsight.

Applicant respectfully notes that in order to support a claim of *prima facie* obviousness, the cited references must teach or suggest each and every element of the invention, and there must be a motivation in the references or the prior art to combine the

references and the prior art as suggested. However, nothing in the art of record, which includes U.S. Publication No. 2002/0118682 to Choe; U.S. Patent No. 5,539,373 to Guha; and U.S. Patent No. 6,157,955 to Narad et al., would teach or suggest, either alone or in combination, a method of performing a search function in an integrated circuit transferring substantially simultaneously a table in parallel from table memory to operational plane memory in the integrated circuit, performing at least one search function on the table in operational plane memory using a search key, and outputting a result of the search function, as now defined by amended Claims 15, 16, 45, and 46.

Applicant respectfully submits that Claims 17-30, which ultimately depend from Claim 16, and Claims 47-60, which ultimately depend from Claim 46 are patentable over the art of record by virtue of their dependency from Claims 1 and 46, respectfully. Further, Applicant submits that Claims 17-30 and 47-60 define patentable subject matter in their own right. Therefore, it is respectfully requested that the rejection of Claims 15-30 and 45-60 under 35 U.S.C. §103(a) be reconsidered and withdrawn.

In view of the foregoing amendment and remarks, entry of the amendments to Claims 15, 16, 45, and 46; favorable consideration of Claims 15, 16, 45, and 46, as amended; favorable reconsideration of Claims 17-30 and 47-60; and allowance of pending claims 15-30 and 45-60 are respectfully and earnestly solicited.

Respectfully submitted,

Rod S. Turner
Registration No.: 38,639
Attorney for Applicant

HOFFMANN & BARON, LLP
6900 Jericho Turnpike
Syosset, New York 11791
(516) 822-3550
RST:mak/me